

# Error Correction in Learning using SVMs

**Srivatsan Laxman**

SLAXMAN@MICROSOFT.COM

*Microsoft Research India  
“Vigyan”, #9, Lavelle Road  
Bangalore 560 001, India*

**Sushil Mittal**

MITTAL@STAT.COLUMBIA.EDU

*Department of Statistics  
Columbia University  
New York, NY 10027, USA*

**Ramarathnam Venkatesan**

VENKIE@MICROSOFT.COM

*Microsoft Research India  
“Vigyan”, #9, Lavelle Road  
Bangalore 560 001, India*

## Abstract

This paper is concerned with learning binary classifiers under adversarial label-noise. We introduce the problem of *error-correction in learning* where the goal is to recover the original clean data from a label-manipulated version of it, given (i) no constraints on the adversary other than an upper-bound on the number of errors, and (ii) some regularity properties for the original data. We present a simple and practical error-correction algorithm called **SubSVMs** that learns individual SVMs on several small-size (log-size), class-balanced, random subsets of the data and then reclassifies the training points using a majority vote. Our analysis reveals the need for the two main ingredients of **SubSVMs**, namely class-balanced sampling and subsampled bagging. Experimental results on synthetic as well as benchmark UCI data demonstrate the effectiveness of our approach. In addition to noise-tolerance, log-size subsampled bagging also yields significant run-time benefits over standard SVMs.

## 1. Introduction

Learning in the presence of noise is notoriously difficult; there are many negative results regarding hardness of learning under adversarial or malicious noise Ben-David et al. (2003); Feldman et al. (2006); Guruswami and Raghavendra (2006); Hastad (1997); Kearns et al. (1994); Long and Servedio (2011), while positive results are mostly known only for the case of random noise or under strong distributional assumptions Blum et al. (1996); Kalai et al. (2008); Sastry et al. (2010); Servedio (2003). Somewhat more encouraging results exist in max-margin settings Buja and Stuetzle (2000); Har-peled et al. (2006); Shalev-Shwartz et al. (2010); Xu et al. (2006) but these methods are computationally prohibitive even for reasonably-sized data.

In this paper, we investigate the learning of binary classifiers under adversarial (worst-case) label-noise. We introduce the problem of *error-correction* in learning, as the task

of correcting the label-errors in training data,  $\hat{\mathbf{D}}$ , given that the original (clean) data,  $\mathbf{D}$ , intrinsically satisfies some regularity properties. (Given negative results such as Guruswami and Raghavendra (2006) regarding the hardness of learning better-than-random hyperplanes even from nearly-separable data, some notion of regularity becomes essential). Informally,  $\mathbf{D}$  is said to be  $r$ -regular if SVMs trained on very small random  $r$ -subsets of  $\mathbf{D}$ , make less than  $\theta$ -fraction errors over all of  $\mathbf{D}$ . We show that every linearly separable  $\mathbf{D}$  exhibits some regularity, and that such a  $\mathbf{D}$  can be recovered from any  $\hat{\mathbf{D}}$  with roughly  $(\frac{1}{2} - 2\theta - O(\log^2 r))$ -fraction of errors. The main idea in our analysis is to apply margin-based generalization bounds under a chosen sampling distribution over  $\mathbf{D}$  and to then adjust the bounds for the noise in  $\hat{\mathbf{D}}$ . To the best of our knowledge, this is the first positive result that is known about learning classifiers under adversarial label-errors.

Our algorithm for error-correction, called **SubSVMs** (Subsample bagging of SVMs) is as follows: Train SVMs on suitably-small, class-balanced, random subsets of  $\hat{\mathbf{D}}$  and reclassify every training point using a simple majority vote. We show that class-balanced sampling over  $\hat{\mathbf{D}}$  minimizes the worst-case probability of drawing less than any-chosen-number of clean points per class from  $\hat{\mathbf{D}}$ . The number of worst-case errors that each SVM in the ensemble makes can grow as the squared-log of the subsample-size used, and this leads us to the final error-correction performance of **SubSVMs**.

In experimental work, we first study the error-correction achievable on synthetic linearly separable data. By comparing against performance under uniform sampling (common in standard bagging) we show that class-balanced sampling plays a vital role in error-correction. Then we show that error-correction based on **SubSVMs** leads to better classifiers which outperform regular SVMs on a range of benchmark data sets from the UCI Machine Learning Repository. Our experiments also clearly demonstrate superiority of **SubSVMs** over regular bagging. We inject high-levels of label-noise in the training data sets (Number of errors was fixed at 75% of the size of the minority class). On previously unseen (clean) test sets, **SubSVMs** even outperformed SVMs that directly used the full test sets for cross-validation. Subsampling at logarithmic sizes also gives **SubSVMs** substantial run-time advantages over standard SVMs and regular bagging.

**Related Work:** Several results show that learning under adversarial noise can be NP-hard Hastad (1997); Kearns et al. (1994); Feldman et al. (2006); Guruswami and Raghavendra (2006). Better results (polynomial-time algorithms) are known in the context of learning max-margin classifiers from noisy data Har-peled et al. (2006); Shalev-Shwartz et al. (2010); Xu et al. (2006). However, these techniques are computationally prohibitive in practice, e.g., the method proposed in Xu et al. (2006) uses SDP solvers that can become impractical even for a hundred training points. Many boosting algorithms, with convex potential functions, have also been shown vulnerable to random classification noise Long and Servedio (2010).

In statistical (rather than adversarial) settings, generalization results for SVMs demonstrate efficient learnability when training and test points are drawn *iid* from the same (even if noisy) distribution Christianini and Shawe-Taylor (2000). Some works have focused on the ineffectiveness of SVMs in the presence of outliers and for noisy class-imbalanced data (e.g., see Akbani et al. (2004); Trafalis and Gilbert (2005); Nath and Bhattacharyya (2007)), albeit without formal analysis. Recently, large-margin half-spaces were shown to be efficiently learnable under small amounts of malicious noise Long and Servedio (2011).

Similarly, Dekel and Shamir (2009) demonstrates learning from multi-teacher data, where a small number of teachers can replace randomly chosen examples arbitrarily. A general framework for distribution-dependent learning in-the-limit was proposed in Caramanis and Mannor (2008); the focus, however, was on establishing informational limits rather than sample complexities. We consider learning under adversarial label-errors given that the original data satisfies some regularity properties. Our error-model is relevant both when the label-errors are inadvertent, whether systematic or random, and when errors are introduced by an adversary explicitly trying to mislead the learning process.

Several studies investigated why (and under what conditions) bagging works by formalizing different notions of stability for predictors and by showing that bagging reduces the variance of unstable predictors (see, e.g., Breiman (1996); Buhlmann and Yu (2002); Elisseff et al. (2005); Grandvalet (2004)). Experimental bias-variance analysis of random aggregation and bagging of SVMs demonstrated that working with small samples achieves greater reduction in the variance component of error than standard bagging (see Valentini (2004)). In another related work, Brodley and Friedl (1999) presented an experimental study of various methods for identifying mislabeled data. All these studies, including the ones that analyze bagging, restricted attention to distribution-based models, rather than adversarial settings.

## 2. Error correction problem in learning

Let  $\mathbf{D} = \{(x_i, y_i) : i = 1, \dots, \ell\}$  be the set of examples in a binary classification problem; the feature vectors,  $x_i$ , come from some domain  $\mathcal{X}$  and the class-labels,  $y_i$ , take values from  $\{-1, +1\}$ . The proportion of minority class points in  $\mathbf{D}$  is denoted  $\beta$ ,  $0 < \beta \leq 0.5$ .

Let  $\Psi_{\mathbf{D}}$  denote a binary SVM classifier trained on  $\mathbf{D}^1$ ; for  $x \in \mathcal{X}$ , the classifier returns the label  $\Psi_{\mathbf{D}}(x) \in \{-1, +1\}$ . We assume that  $\Psi_{\mathbf{D}}$  is suitable for the given classification task. However,  $\mathbf{D}$  is not available to train the learning algorithm. Instead, the learner only has access to  $\hat{\mathbf{D}} = \{(x_i, \hat{y}_i) : i = 1, \dots, \ell\}$ , which is a *label-manipulated* version of  $\mathbf{D}^2$ .

The adversary is allowed to flip labels of *no more than*  $\rho\beta\ell$  examples in  $\mathbf{D}$ , where  $\rho$  is referred to as the *error parameter*. Since we place no other restrictions on the points the adversary can manipulate, we must have the constraint  $0 \leq \rho < 1$  (otherwise, we may be left with no training examples for one class).

The error-correction problem is concerned with recovering the original clean data  $\mathbf{D}$  (or a close approximation of it) from its label-manipulated version  $\hat{\mathbf{D}}$ . To this end, we will allow some ‘regularity’ assumptions on the original data  $\mathbf{D}$ , which essentially guarantee that SVMs trained on sufficiently-small random subsets of  $\mathbf{D}$  can classify the points in  $\mathbf{D}$  with high accuracy. Regularity is an intrinsic property of the original data, which can manifest and be measured in many ways; one way is to measure the redundancy structure exposed by the quadratic program underlying the max-margin formulation of SVMs.

**Definition 1 (Data Regularity)** *Let  $\mathcal{D}_*$  be any (discrete) probability distribution over  $\mathbf{D}$  and let  $\mathbf{S} \sim \mathcal{D}_*$ ,  $|\mathbf{S}| \geq r$ , denote a collection of points drawn iid from  $\mathcal{D}_*$ . For any  $\delta < 0.5$  and  $\theta < 0.5$ ,  $\mathbf{D}$  is said to be  $r$ -regular at  $(\delta, \theta)$  if with probability at least  $1 - \delta$  over choice*

---

1.  $\Psi_{\mathbf{S}}$  denotes the SVM trained on  $\mathbf{S}$ , etc.  
 2.  $\hat{\mathbf{D}}$  is also referred to as the *corrupted* or *noisy* data.

of  $\mathbf{S}$ , the expected error-rate of  $\Psi_{\mathbf{S}}$  does not  $\theta$  with respect to test examples also drawn iid from  $\mathcal{D}_*$ .

We are interested in regularity at small  $r$ , such as at  $O(\log \ell)$  or  $O(\log^2 \ell)$ . Data regularity can be thought of as a measure of *redundancy* needed to admit learning in the presence of adversarial label-noise. This is, in a sense, akin to the redundancy encoded into a message for enabling error-correction in coding theory. Regularity is a simple property that is satisfied by data from which good binary classifiers can be easily learnt, e.g., every linearly separable data set is regular.

**Lemma 2 (Separability implies Regularity)** *Consider any linearly separable  $\mathbf{D}$  with margin  $\gamma$ . For any fixed  $\delta < 0.5$  and  $\theta < 0.5$ , there exists  $r \in \mathbb{Z}^+$  such that  $\mathbf{D}$  is  $r$ -regular at  $(\delta, \theta)$ .*

The proof makes use of the following 2-norm soft-margin bound from SVM generalization theory Christianini and Shawe-Taylor (2000):

**Theorem 3** (Christianini and Shawe-Taylor, 2000, Theorem 4.22) *Consider thresholding real-valued linear functions  $\mathcal{L}$  with unit weight vectors on an inner product space  $\mathcal{X}$  and fix  $\gamma \in \mathbb{R}^+$ . There is a constant  $c$ , such that for any probability distribution  $\mathcal{D}$  on  $\mathcal{X} \times \{-1, +1\}$  with support in a ball of radius  $R$  around the origin, with probability  $1 - \delta$  over  $\ell$  random (training) examples  $\mathbf{D} = \{(x_1, y_1), \dots, (x_\ell, y_\ell)\}$ , any hypothesis  $f \in \mathcal{L}$  has error no more than*

$$\Pr_{(x,y) \sim \mathcal{D}} [f(x) \neq y] \leq \frac{c}{\ell} \left( \frac{R^2 + \|\boldsymbol{\xi}\|_2^2}{\gamma^2} \log^2 \ell + \log \frac{1}{\delta} \right) \quad (1)$$

where  $\boldsymbol{\xi} = (\xi_1, \dots, \xi_\ell)$  is the margin slack vector with respect to  $f$  and  $\gamma$ . The entries of  $\boldsymbol{\xi}$  are fixed as follows:  $\xi_i = \max(0, \gamma - y_i f(x_i))$ ,  $i = 1, \dots, \ell$ .

Since  $\mathbf{D}$  is separable with margin  $\gamma$ , every subset of  $\mathbf{D}$  is also separable with margin at least  $\gamma$ . Thus, the max-margin separator of every subset of  $\mathbf{D}$  will have margin slack vector  $\boldsymbol{\xi} = 0$  (with respect to the chosen subset). Fixing  $\mathcal{D} = \mathcal{D}_*$  in Theorem 3, the generalization error of  $\Psi_{\mathbf{S}}$  trained on any  $\mathbf{S} \sim \mathcal{D}_*$ ,  $|\mathbf{S}| = r$ , is given by

$$\Pr_{(x,y) \sim \mathcal{D}_*} [\Psi_{\mathbf{S}}(x) \neq y] \leq \frac{c}{r} \left( \frac{R^2}{\gamma^2} \log^2 r + \log \frac{1}{\delta} \right) \quad (2)$$

Lemma 2 follows since the RHS of (2) is  $O(\log^2 r/r)$ .

**Definition 4 (Error-correction in Learning)** *Given that  $\mathbf{D}$  and  $\hat{\mathbf{D}}$  disagree on no more than  $\rho\beta$ -fraction of labels, and given that  $\mathbf{D}$  satisfies some regularity properties, the problem of error-correction in learning is to recover a data set  $\tilde{\mathbf{D}}$  with as few label disagreements with  $\mathbf{D}$  as possible.*

We make no assumptions regarding the nature of label-errors (such as if they are statistical or otherwise), or regarding the separate values of error-parameter ( $\rho$ ) and true fraction of minority-class ( $\beta$ ); we are only given that the total fraction of label-errors does not exceed  $\rho\beta$ ,  $0 \leq \rho < 1$  and  $0 < \beta \leq 0.5$ .

---

**Algorithm 1** [SubSVMs] Subsampled bagging of SVMs
 

---

**Input:** Corrupted data  $\widehat{\mathbf{D}} = \{(x_1, \widehat{y}_1), \dots, (x_\ell, \widehat{y}_\ell)\}$ ; size,  $s$ , of subsample; sampling bias  $p$ ; number of SVMs  $J$  (typically,  $p = \frac{1}{2}$  and  $s = \log \ell$  or  $s = \log^2 \ell$ )

**Output:** Error-corrected data  $\widetilde{\mathbf{D}} = \{(x_1, \widetilde{y}_1), \dots, (x_\ell, \widetilde{y}_\ell)\}$

---

```

/* Training */
for  $j = 1$  to  $J$  do
    Draw random subset  $\widehat{\mathbf{S}}_j \sim \mathcal{D}_{p\widehat{\mathbf{D}}}$  of size  $|\widehat{\mathbf{S}}_j| = s$ 
    Train SVM  $\Psi_{\widehat{\mathbf{S}}_j}$ 

/* Error-correction */
for  $i = 1$  to  $\ell$  do
    Set  $\widetilde{y}_i$  to the majority label in  $\{\Psi_{\widehat{\mathbf{S}}_1}(x_i), \dots, \Psi_{\widehat{\mathbf{S}}_J}(x_i)\}$ 
Output  $\widetilde{\mathbf{D}} = \{(x_1, \widetilde{y}_1), \dots, (x_\ell, \widetilde{y}_\ell)\}$ 
    
```

---

### 3. The SubSVMs algorithm

We first define a key ingredient of the SubSVMs algorithm that we refer to as *p-biased sampling*.

**Definition 5** (*p-biased Sampling*) *The process of p-biased sampling of  $\widehat{\mathbf{D}}$  refers to the following two steps, executed in the stated order: (1) choose the minority class<sup>3</sup> of  $\widehat{\mathbf{D}}$  with probability  $p$  (or the other class with probability  $1 - p$ ) and (2) pick a point uniformly at random from the restriction of  $\widehat{\mathbf{D}}$  to the chosen class. The corresponding sampling distribution is denoted  $\mathcal{D}_{p\widehat{\mathbf{D}}}$  and  $\widehat{\mathbf{S}} \sim \mathcal{D}_{p\widehat{\mathbf{D}}}$  denotes that  $\widehat{\mathbf{S}}$  is a random collection of points drawn iid with respect to  $\mathcal{D}_{p\widehat{\mathbf{D}}}$ .*

The case of  $p = 0.5$  is referred to as *class-balanced sampling* of  $\widehat{\mathbf{D}}$ ; if  $\widehat{\beta}$  denotes the fraction of minority class points in  $\widehat{\mathbf{D}}$ , the case of  $p = \widehat{\beta}$  is equivalent to *uniform sampling* over  $\widehat{\mathbf{D}}$ .

Algorithm 1 lists the pseudo-code for *subsampled bagging of SVMs* (SubSVMs). Our analysis (in Secs. 3.1-3.2) reveals two important aspects of SubSVMs:

- Class-balanced sampling provides optimal protection against worst-case label-errors.
- The fraction of errors that can be tolerated ( $\rho\beta$ ) reduces as the squared-log of sample-size  $s$ .

Based on the above, we use class-balanced sampling ( $p = 1/2$ ) and choose  $s$  to be  $\log \ell$  or  $\log^2 \ell$ .

#### 3.1 Error correction analysis

Our analysis uses the margin-based generalization bound for SVMs with respect to a sampling distribution over the original (clean) data  $\mathbf{D}$  and then adjusts the bound to accommodate the number of label-errors in the corrupted training set  $\widehat{\mathbf{D}}$ .

---

3. If both classes of  $\widehat{\mathbf{D}}$  are of identical size, one of them is arbitrarily fixed as the ‘minority class’.

Consider the general case of *Algorithm 1*, where the random subsets  $\widehat{\mathbf{S}}_j$  are drawn *iid* from  $\mathcal{D}_{p\widehat{\mathbf{D}}}$ . Let  $\mathbf{D}$  be linearly separable with margin  $\gamma$ . Consider a set of points  $\widehat{\mathbf{S}} \sim \mathcal{D}_{p\widehat{\mathbf{D}}}$ . We now need to compute the expected error-rate of  $\Psi_{\widehat{\mathbf{S}}}$  with respect to test points drawn uniformly from  $\mathbf{D}$  (This is the main quantity of interest in the error-correction setting). For this, we first compute the expected error-rate  $\epsilon$  when the training and test cases are both drawn *iid* from  $\mathcal{D}_{p\widehat{\mathbf{D}}}$ . This is done by using *Theorem 3* (Christianini and Shawe-Taylor, 2000, Theorem 4.22) with  $f = \Psi_{\widehat{\mathbf{S}}}$  and  $\mathcal{D} = \mathcal{D}_{p\widehat{\mathbf{D}}}$  (See next paragraph for details). The error-rate can at most become  $\epsilon/p^*$ , where  $p^* = \min\{p, 1 - p\}$ , when considering test cases drawn *uniformly* from  $\widehat{\mathbf{D}}^4$ . Finally, in any uniformly drawn sample from  $\widehat{\mathbf{D}}$ , the expected fraction of label disagreements with respect to the corresponding points in  $\mathbf{D}$  is  $\rho\beta$ . Hence, the desired expected error-rate of  $\Psi_{\widehat{\mathbf{S}}}$ , where  $\widehat{\mathbf{S}} \sim \mathcal{D}_{p\widehat{\mathbf{D}}}$  but the test points are drawn uniformly from  $\mathbf{D}$ , is given by  $\epsilon/p^* + \rho\beta$ .

We now return to the computation of error-rate  $\epsilon$  when train and test points are both drawn *iid* from  $\mathcal{D}_{p\widehat{\mathbf{D}}}$ . Whenever  $\widehat{\mathbf{S}}$  contains *at least*  $r/2$  clean points per class, the SVM of the corresponding  $r$ -size (clean) subset of  $\widehat{\mathbf{S}}$  would make no more than  $(s - r)$  mistakes on the rest of  $\widehat{\mathbf{S}}$ . Each of these mistakes would be no farther than  $2R$  from either supporting hyperplane. Also, the margin of this SVM would be at least  $\gamma$  (the max-margin achieved on the whole of  $\mathbf{D}$ ). The 2-norm SVM objective has the same form as the error-bound in (1). Hence, we apply *Theorem 3* with  $\|\xi\|_2^2 = 4R^2(s - r)$  and with margin  $\gamma$ , to obtain the generalization bound,  $\epsilon$ . If  $\eta$  is an upper-bound on the probability that  $\widehat{\mathbf{S}}$  contains *less than*  $r/2$  clean points from either class, then with probability at least  $(1 - \eta - \delta)$

$$\Pr_{(x,y) \sim \mathcal{D}_{p\widehat{\mathbf{D}}}} [\Psi_{\widehat{\mathbf{S}}}(x) \neq y] \leq \frac{c}{s} \left( \frac{R^2 + 4R^2(s - r)}{\gamma^2} \log^2 s + \log \frac{1}{\delta} \right) \stackrel{\text{def}}{=} \epsilon.$$

Recall that this error-rate,  $\epsilon$ , over test points drawn from  $\mathcal{D}_{p\widehat{\mathbf{D}}}$ , translates to an error-rate of  $\epsilon/p^* + \rho\beta$  for test points drawn uniformly over  $\mathbf{D}$ . Thus, the final expression for probability of error of  $\Psi_{\widehat{\mathbf{S}}}$  with respect to test points drawn uniformly from  $\mathbf{D}$ , denoted  $\varphi$ , can be written as follows:

$$\Pr[\Psi_{\widehat{\mathbf{S}}}(x) \neq y] \leq (1 - \eta - \delta) \left[ \frac{\epsilon}{p^*} + \rho\beta \right] + \eta + \delta \stackrel{\text{def}}{=} \varphi.$$

We use  $J$  SVMs based on  $J$  random sets such as  $\widehat{\mathbf{S}}$ . Thus, if  $\varphi < 0.5$ , then (by Hoeffding Inequality Hoeffding (1963)) the probability of a majority vote making a mistake with respect to  $\mathbf{D}$  cannot exceed  $\exp[-2J(0.5 - \varphi)^2]$ . This gives us error-correction (in the sense that  $\mathbf{D}$  can be correctly recovered from  $\widehat{\mathbf{D}}$ ). To enforce the condition  $\varphi < 0.5$ , we must have  $\rho\beta < 1 - \epsilon/p^* - [2(1 - \eta - \delta)]^{-1}$ . Finally, if  $\mathbf{D}$  is  $r$ -regular at  $(\delta, \theta)$ , then we have

$$\begin{aligned} \epsilon &= \frac{c}{s} \left( \frac{R^2}{\gamma^2} \log^2 s + \log \frac{1}{\delta} \right) + \frac{c}{s} \left( \frac{4R^2(s - r)}{\gamma^2} \log^2 s \right) \\ &\leq \theta + \frac{c}{s} \left( \frac{4R^2(s - r)}{\gamma^2} \log^2 s \right). \end{aligned} \tag{3}$$

This leads to our main result about **SubSVMs**:

---

4. See Appendix A for a short proof.

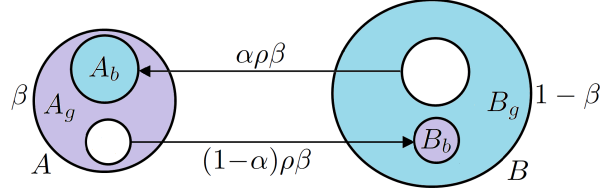


Figure 1: The data corruption process. Let  $A$  be the minority class with  $\beta$ -fraction of points in  $\mathbf{D}$ .  $A_b$  represents the  $\alpha$ -fraction of corrupted points, originally in class- $B$ , but wrongly assigned to class- $A$  in  $\hat{\mathbf{D}}$ ; similarly  $B_b$  represents the class- $A$  points in  $\mathbf{D}$  that were mislabeled as class- $B$  in  $\hat{\mathbf{D}}$ . A total of  $\rho\beta$ -fraction of points are corrupted in  $\hat{\mathbf{D}}$ .

**Theorem 6 (Error-correction)** Consider linearly separable  $\mathbf{D}$  with margin  $\gamma$  and  $\beta$ -fraction of minority-class points. Fix  $\delta < 0.5$  and let  $\mathbf{D}$  be  $r$ -regular at  $(\delta, \theta)$ . Consider  $\hat{\mathbf{D}}$  with error-rate  $\rho$  and  $\hat{\mathbf{S}} \sim \mathcal{D}_{p\hat{\mathbf{D}}}$ ,  $|\hat{\mathbf{S}}| = s$ . Let  $\Pr[\hat{\mathbf{S}} \text{ contains } < r/2 \text{ clean points per class}] \leq \eta$ . If the number of label-errors in  $\hat{\mathbf{D}}$  is bounded by

$$\rho\beta < 1 - 2\theta - \left[ \frac{1}{2(1 - \eta - \delta)} + \frac{4R^2 c(s - r) \log^2 s}{\gamma^2 s} \right] \quad (4)$$

where  $R$  denotes the radius of the ball enclosing the data and  $c$  is the constant from Theorem 3, then the probability of error for SubSVMs with respect to points drawn uniformly from  $\mathbf{D}$  is at most  $\exp[-2J(0.5 - \varphi)^2]$ , where  $\varphi = \eta + \delta + (1 - \eta - \delta)[\epsilon/p^* + \rho\beta]$  and  $p^* = \min\{p, 1 - p\}$ .

Hence, perfect error-correction is attained for  $\varphi < 0.5$ .

### 3.2 Importance of Class-balanced Sampling

The bound in (4) has two groups of parameters. In the first group, we have  $r$ ,  $\delta$  and  $\theta$ , which are fixed by the regularity properties of  $\mathbf{D}$ . In the second group, we have  $s$  and  $\eta$ , which are both determined by our sampling strategy. Since  $\eta$  depends on the sampling bias  $p$ , we now discuss how to fix  $p$  and  $s$  for optimal error-correction performance.

From (4) it is clear that, to maximize the number of errors that can be tolerated, we must minimize the quantity in square brackets. The first term inside the brackets is minimized when  $\eta$  is minimum. Fig. 1 provides a graphical depiction of the data corruption process. The optimal value of  $\eta$  typically depends on the direction-of-attack parameter,  $\alpha$ , the error parameter  $\rho$ , and the true size,  $\beta$ , of the minority class in  $\mathbf{D}$ . However, neither of these is known to the learner; only an upper-bound on the fraction of label-errors in  $\hat{\mathbf{D}}$  is known. So we design our algorithm to limit the impact of worst-case label-errors. Specifically, we choose  $p = 0.5$  since it minimizes  $\eta$  in a manner that is agnostic to the true values of  $\alpha$ ,  $\rho$  and  $\beta$ . We state this formally in Lemma 7 below.

**Lemma 7 (Class-balanced Sampling)** Fix any  $r \in \mathbb{Z}^+$ . Given  $\mathbf{D}$  with  $\beta$ -fraction of minority-class points and  $\hat{\mathbf{D}}$  with at most  $\rho\beta$ -fraction label-errors w.r.t.  $\mathbf{D}$ , class-balanced

sampling of  $\widehat{\mathbf{D}}$  minimizes a worst-case upper-bound on  $\eta$  (probability that the sample drawn contains less than  $r/2$  clean points per class) if the size,  $s$  ( $\geq r$ ), of the sample satisfies

$$s \geq 2r + 4 \left( r \log 2 + \log^2 2 - \log 4 \right)^{\frac{1}{2}} + \log 16 - 4 \quad (5)$$

The main intuition behind the proof is that, in the absence of any specific information regarding  $\rho$ ,  $\beta$  and  $\alpha$ , choosing the sampling bias  $p$  on either side of 0.5 is vulnerable to one of the attack directions, thereby increasing the worst-case value of  $\eta$ . (See Appendix B for the proof).

The second term inside the square brackets of (4) is smallest (and equal to zero) for  $s = r$ . However, Lemma 7 shows that this is not optimal for  $\eta$ , since  $s = r$  fails the condition in (5). In fact, for smaller  $s$ ,  $\eta$  may even be maximized at  $p = 0.5$ ; in general, the minimizer of  $\eta$  will no longer be agnostic to  $\rho$ ,  $\beta$  and  $\alpha$ . However, when  $s$  is set to the lower-bound of (5), the second term inside square brackets of (4) becomes  $O(\log^2 r)$ . This gives us our next lemma.

**Lemma 8 (Subsampled Bagging)** *Let  $\mathbf{D}$  be linearly separable and  $r$ -regular at  $(\delta, \theta)$  and let  $\widehat{\mathbf{D}}$  contain at most  $(\rho\beta)$ -fraction of adversarial label-errors. SubSVMs based on class-balanced sampling and with subsample-size,  $s$ , set to the lower-bound in (5), can perfectly recover the original  $\mathbf{D}$ , provided the fraction of label-errors in  $\widehat{\mathbf{D}}$  is bounded above as follows:*

$$\rho\beta < 1 - 2\theta - \left[ \frac{1}{2(1 - \eta - \delta)} + O(\log^2 r) \right] \quad (6)$$

Since the above lemma requires  $s$  to be set at the lower-bound of (5) it might appear that we are operating on a knife-edge for choosing the subsample size. Luckily, this is not the case, because if the data is regular at  $r$ , it would also be regular with same  $\theta$  for every  $r' > r$ . Hence, we could set  $s$  to the lower bound in (5) corresponding to  $r'$  and the above Lemma would still hold, though with  $O(\log^2 r')$  rather than  $O(\log^2 r)$  inside the square brackets. As a result, the number of worst-case errors allowed reduces for  $r' > r$  and this is the reason why we use *subsampled* bagging. Typically, we choose  $s$  to be  $\log \ell$  or  $\log^2 \ell$  (rather than  $\ell$ , which is the usual case in bagging). As long as the data is  $r$ -regular for some  $r < s$  that satisfies (5) SubSVMs will give us error-correction. As a side-benefit subsampling at logarithmic sizes will give us dramatic run-time advantages over regular SVMs. Our experimental results clearly demonstrate this aspect of SubSVMs.

## 4. Experiments

We present experimental results of SubSVMs on simulated, linearly separable data as well as LIBSVM extracts of some UCI data sets<sup>5</sup>. SVMs are known to perform well on these data sets, so they can play the role of clean data in our experiments.

Our data corruption process follows Fig. 1. Given ‘clean’ training data  $\mathbf{D}$  of size  $\ell$  with minority class of size  $\beta\ell$ ,  $0 < \beta \leq 0.5$ , the parameters  $\rho$  and  $\alpha$  control the corruption. We randomly pick  $\rho\beta\ell$  points for corruption, of which,  $\alpha$ -fraction are picked uniformly at

5. <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets>



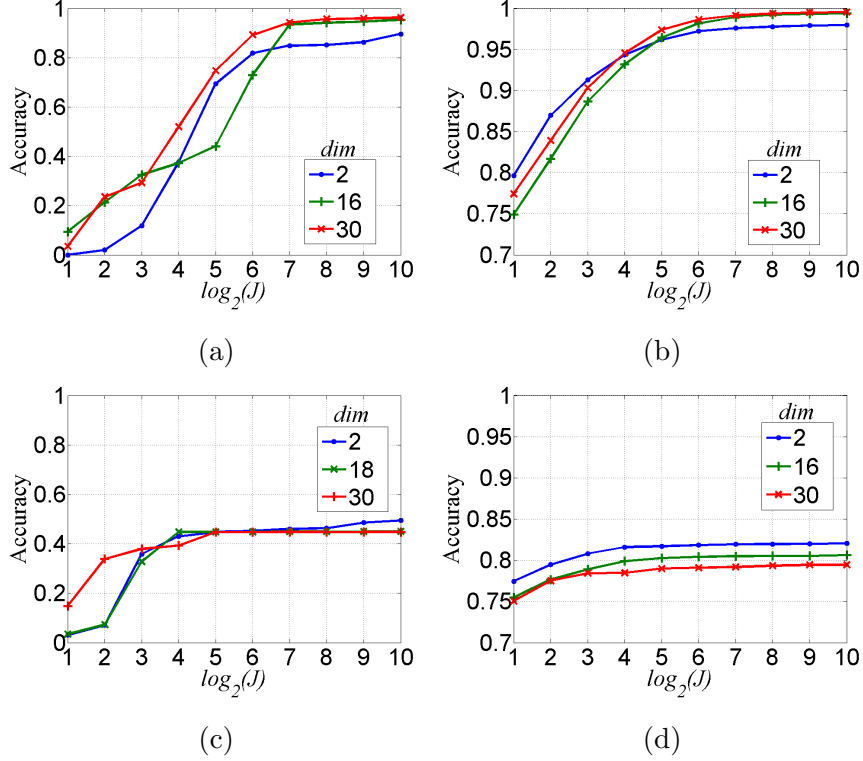


Figure 2: Importance of class-balanced sampling ( $p = 1/2$ ) in *Algorithm 1*. (a) Worst-case and (b) average-case accuracy obtained using class-balanced sampling ( $p = 1/2$ ) over different label-manipulated data sets as a function of  $J$ . (c) Worst-case and (d) average-case accuracy obtained using SubSVMs with uniform sampling ( $p = \beta$ ).

random from the minority class and  $(1 - \alpha)$ -fraction from the other. By varying the *attack direction*  $\alpha$ , we generated a wide range of corrupted data with different degrees of difficulty for binary classification.

#### 4.1 Synthetic Data Experiments

In the first experiment, we generated ‘clean’ data sets  $\mathbf{D}$  comprising of 1000  $d$ -dimensional data points from a mixture of two Gaussian distributions, each with a covariance of  $0.1\mathbf{I}_d$  and a distance of two units between means. Three values of  $d$  were used: 2, 16 and 30. A constant margin of 0.2 units was enforced and misclassified points were manually removed. The value of  $\beta$  was varied between  $[0.05, 0.5]$  in steps of 0.05,  $\rho = 0.75$  and  $\alpha$  was varied between  $[0.0, 1.0]$  in steps of 0.25.

We studied the importance of class-balanced sampling in *Algorithm 1* (SubSVMs) by comparing two versions of it - one with class-balanced sampling ( $p = 1/2$ ) and the other with uniform sampling ( $p = \beta$ ). For every  $d$ , the data corresponding to each  $[\beta, \alpha]$  pair was subjected to 10 random corruptions. Figs. 2a and 2b summarize the results for class-

Data set	Feature dimension	Training set		Test set	
		Total size	Size of minority class	Total size	Size of minority class
a1a	123	1605	395 (25%)	30956	7446 (24%)
a2a		2265	572 (25%)	30296	7269 (24%)
a3a		3185	773 (24%)	29376	7068 (24%)
a4a		4781	1188 (25%)	27780	6653 (24%)
a5a		6414	1569 (25%)	26147	6272 (24%)
splice	60	1000	483 (48%)	2175	1044 (48%)
mushrooms	112	6093	2937 (48%)	2031	979 (48%)
svmguidel	4	3089	1089 (35%)	4000	2000 (50%)
w1a	300	2477	72 (3%)	47272	1407 (3%)
w2a		3470	107 (3%)	46279	1372 (3%)
w3a		4912	143 (3%)	44837	1336 (3%)
w4a		7366	216 (3%)	42383	1263 (3%)
w5a		9888	281 (3%)	39861	1198 (3%)

Table 1: LIBSVM UCI data extracts and their characteristics.

balanced sampling and Figs. 2c and 2d for uniform sampling. As expected, based on Theorem 6, the number of mistakes made decays exponentially with increasing  $J$ . Near-perfect error-correction is achieved using  $p = 1/2$  for  $J$  as small as  $2^7$ . For  $p = \beta$ , the worst-case and average-case performances are worse by about 60% and 20%, respectively. This experimentally validates Lemma 7 for using class-balanced sampling in **SubSVMs**.

## 4.2 UCI Data Experiments

We now report the performance of **SubSVMs** on held-out test data using the LIBSVM UCI extracts. There can be two ways to test this, either the error-corrected training data can be used to retrain a fresh standard SVM or we can just use majority voting over the  $J$  SVMs already trained in **SubSVMs**. In our experiments, both these approaches yielded very similar results. Therefore, we avoid retraining cost and report results using the majority voting method.

Table 1 shows the data characteristics of the 13 data sets used. The fraction of the minority class,  $\beta$  ranges from 0.03 to 0.48 in training sets and from 0.03 to 0.50 in test sets. Also, the feature dimension varies between 4 to 300. Note that although these data sets are not linearly separable, they are still referred to as ‘clean’ before they are subjected to label-manipulation. For generating different types of attacks,  $\rho = 0.75$  was used while the value of  $\alpha$  was varied between  $[0.0, 1.0]$  in steps of 0.25. We compare **SubSVMs** against of four other SVM-based classifiers:

1. **Oracle-SVM**: Standard SVM learnt over training data with parameters fixed by cross-validating directly over *clean* test set.
2. **Blind-SVM**: Standard SVM learnt over training data with parameters fixed based on the best average performance over *all* test sets. This is similar to **Oracle-SVM**, except

that a single set of parameters is used for all data sets. This helps assess the feasibility of blindly fixing the same set of parameters for all test sets.

3. **Bag-SVM**: Regular bagging of SVMs where each SVM in the ensemble is trained on a bootstrap sample of size same as the original data (sampled with replacement). All SVMs use the same set of optimum parameters, which were determined through test set cross-validation of **Oracle-SVM**.
4. **CV-SVM**: Standard SVM with parameters chosen through four-fold cross-validation on the training data. In all the experiments, the results of **CV-SVM** are averaged over five different random splits of the training data for cross-validation.

All cross-validations were performed by varying the penalty parameter  $C$  between 1 and 100, ratio of the weights of the two classes  $W$  between 0.1 and 10 and the RBF kernel parameter  $\sigma^2$  between  $0.1/d$  and  $10/d$ , where  $d$  is the data dimensionality. For **SubSVMs**, the values of  $C = 100$ ,  $w = 1$ ,  $\sigma^2 = 1/d$ ,  $s = \log^2 \ell$  and  $J = 1000$  were fixed for all data sets without performing any sort of cross-validation. All the SVMs were trained under L-2 loss, although similar results were also obtained under L-1 loss.

Note that **Oracle-SVM**, **Blind-SVM** and **Bag-SVM** use information about test set labels to obtain their corresponding set of optimum parameters for training. This gives them an unfair advantage over **CV-SVM** and **SubSVMs** that are both agnostic to test set labels.

**Performance measure**: The UCI data sets exhibit a wide range of class imbalance - a1a-a5a are moderately imbalanced, splice, mushrooms and svmguide1 are class-balanced while w1a-w5a are highly imbalanced. For imbalanced data, high accuracies can be trivially achieved by labeling all points with the majority class label. Since accuracy is ineffective in such settings, we use its skew-insensitive version called Balanced Accuracy<sup>6</sup> (BAC) Brodersen et al. (2010). Note that for class-balanced data, BAC reduces to accuracy.

Table 2 summarizes the results of all the five methods on clean as well as corrupted versions of the data. For every data set, 10 random corruptions were performed w.r.t. the corresponding attack direction  $\alpha$  and the averaged results are reported. Winning results, when significantly better than the rest, are highlighted<sup>7</sup>.

- **SubSVMs** is almost always significantly better than all the other methods (by 5% or more) and is never significantly worse. The advantage of **SubSVMs** is visible in both balanced and imbalanced data; for imbalanced data, the advantage increases for smaller  $\alpha$ . This is because the quality of minority-class data falls sharply with  $\alpha$ .
- **Oracle-SVM** is at least as good as **Blind-SVM**. This is because **Oracle-SVM** tunes parameters individually for each test set, while **Blind-SVM** fixes the same parameters across all test sets.
- **Oracle-SVM**, **Blind-SVM** and **Bag-SVM** are better than **CV-SVM**. This is because all three methods cross-validate directly on the test sets.

6. See Appendix C for details of this measure.

7. Std. devs. were negligible (mostly  $< 0.02$ , max 0.06).

	<i>Method</i>	a1a	a2a	a3a	a4a	a5a	splc	mush	svm1	w1a	w2a	w3a	w4a	w5a
<i>clean</i>	Oracle-SVM	0.76	0.77	0.76	0.76	0.76	0.91	1.00	0.97	0.75	0.76	0.78	0.79	0.81
	Blind-SVM	0.75	0.75	0.75	0.76	0.76	0.89	1.00	0.97	0.75	0.76	0.78	0.79	0.81
	Bag-SVM	0.76	0.77	0.76	0.76	0.76	0.90	1.00	0.97	0.73	0.75	0.77	0.79	0.79
	CV-SVM	0.76	0.76	0.76	0.76	0.76	0.90	1.00	0.97	0.70	0.69	0.71	0.72	0.73
	SubSVMs	<b>0.81</b>	0.81	<b>0.81</b>	<b>0.81</b>	<b>0.81</b>	0.86	0.98	0.96	<b>0.85</b>	<b>0.86</b>	<b>0.86</b>	<b>0.87</b>	<b>0.88</b>
$\alpha = 1.0$	Oracle-SVM	0.79	0.80	0.80	0.81	0.81	0.57	0.64	0.89	0.74	0.77	0.78	0.79	0.81
	Blind-SVM	0.74	0.74	0.75	0.77	0.77	0.57	0.54	0.84	0.74	0.76	0.77	0.79	0.80
	Bag-SVM	0.79	0.80	0.80	0.81	0.81	0.55	0.62	0.89	0.73	0.76	0.78	0.79	0.80
	CV-SVM	0.78	0.78	0.79	0.79	0.79	0.54	0.54	0.89	0.67	0.69	0.69	0.73	0.74
	SubSVMs	0.79	0.79	0.79	0.80	0.80	<b>0.77</b>	<b>0.97</b>	0.90	<b>0.79</b>	<b>0.82</b>	<b>0.83</b>	<b>0.85</b>	0.85
$\alpha = 0.5$	Oracle-SVM	0.63	0.63	0.63	0.64	0.64	0.73	0.99	0.90	0.64	0.65	0.67	0.68	0.69
	Blind-SVM	0.63	0.63	0.63	0.64	0.64	0.62	0.85	0.88	0.64	0.64	0.65	0.66	0.66
	Bag-SVM	0.64	0.64	0.64	0.64	0.64	0.73	0.99	0.90	0.63	0.64	0.66	0.67	0.68
	CV-SVM	0.63	0.63	0.63	0.64	0.64	0.70	0.94	0.90	0.59	0.56	0.58	0.59	0.58
	SubSVMs	<b>0.79</b>	<b>0.79</b>	<b>0.79</b>	<b>0.80</b>	<b>0.80</b>	0.74	0.98	0.93	<b>0.78</b>	<b>0.80</b>	<b>0.81</b>	<b>0.84</b>	<b>0.85</b>
$\alpha = 0.0$	Oracle-SVM	0.55	0.55	0.55	0.55	0.55	0.55	0.62	0.50	0.54	0.55	0.55	0.55	0.55
	Blind-SVM	0.55	0.55	0.55	0.55	0.55	0.53	0.62	0.50	0.54	0.55	0.55	0.55	0.55
	Bag-SVM	0.53	0.53	0.54	0.54	0.54	0.54	0.59	0.50	0.53	0.54	0.54	0.54	0.54
	CV-SVM	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.53	0.52	0.52	0.51	0.51
	SubSVMs	<b>0.78</b>	<b>0.79</b>	<b>0.80</b>	<b>0.80</b>	<b>0.80</b>	<b>0.79</b>	<b>0.98</b>	<b>0.96</b>	<b>0.74</b>	<b>0.78</b>	<b>0.81</b>	<b>0.82</b>	<b>0.83</b>

Table 2: Balanced Accuracy (BAC) under L-2 loss for clean and noisy versions of UCI data sets. For different types of attacks (different  $\alpha$ ) the results for each method are averaged over 10 different noisy versions. ‘splc’, ‘mush’ and ‘svm1’ stand for splice, mushrooms and svmguide1. Only CV-SVM and SubSVMs are agnostic to the true test labels. The cases where one of the methods is significantly better than all others ( $\geq 0.05$ ) are highlighted.

- Bag-SVM’s performance is similar to that of Oracle-SVM. This is consistent with Valentini (2004) that also reported no benefit in bagging SVMs (since SVMs are stable classifiers).
- CV-SVM is the worst performing method and is often significantly worse than others<sup>8</sup>. This shows its ineffectiveness under noisy settings.

Similar results were also obtained using Skew-Insensitive F-score (SIF) Flach (2003). Results using Area Under the Curve (AUC) and accuracy, their unsuitability for imbalanced data notwithstanding, are reported in Appendix D.

8. The case of clean, balanced data is the only exception.

<i>Method</i>	a1a	a2a	a3a	a4a	a5a	splc	mush	svm1	w1a	w2a	w3a	w4a	w5a
Oracle-SVM	218	318	456	721	1432	43	561	44	256	374	522	886	1433
Blind-SVM	216	314	450	718	1420	43	558	41	254	368	514	869	1416
Bag-SVM	65	145	293	594	2287	63	1546	208	35	72	132	351	839
CV-SVM	170	351	715	1804	3501	202	3276	363	95	196	384	915	1688
SubSVMs	3	4	4	5	6	5	8	4	4	5	4	5	6

Table 3: Training times in seconds (rounded to the closest integer) for all the methods trained using L-2 loss averaged over different corruptions corresponding to the results presented in Table 2. Note that for **SubSVMs**, the reported time is the time taken to train all the  $J = 1000$  SVMs on  $s = \log^2 \ell$ -size subsets.

**Run-times:** Table 3 summarizes training times averaged over different types of attacks. **SubSVMs** is clearly much faster than all other methods<sup>9</sup>. While our experiments were based on single-core implementations, **SubSVMs** can be easily parallelized to handle very large-scale problems.

## 5. Conclusions

We present a simple algorithm (**SubSVMs**) for learning binary classifiers under adversarial label-noise. **SubSVMs** can efficiently correct a bounded number of adversarial label-errors introduced in linearly separable data. Extensions to handle attribute noise and multi-class settings are important directions for future work. It would also be interesting to explore applicability of **SubSVMs** for solving large, noisy, real-world problems, where SVMs typically perform poorly.

---

9. See Appendix D for more detailed run-times.

## Appendix A. Error-rate of $\Psi_{\hat{\mathbf{S}}}$ w.r.t. samples drawn uniformly from $\hat{\mathbf{D}}$

Let  $\epsilon_1$  and  $\epsilon_2$  be the class conditional error rates for the two classes. Without loss of generality let  $\epsilon_2 \geq \epsilon_1$ . In the absence of the knowledge whether  $\epsilon_2$  is associated with the minority class or the majority class, the overall error rate of  $\Psi_{\hat{\mathbf{S}}}$  w.r.t. samples drawn *iid* from  $\mathcal{D}_{p\hat{\mathbf{D}}}$  is given by

$$\epsilon = \max(p\epsilon_1 + (1-p)\epsilon_2, (1-p)\epsilon_1 + p\epsilon_2) \leq \epsilon_2. \quad (7)$$

Therefore, if  $\epsilon = p\epsilon_1 + (1-p)\epsilon_2$ , then

$$\epsilon_2 = \frac{\epsilon - p\epsilon_1}{1-p} \leq \frac{\epsilon}{1-p} \quad (8)$$

and if  $\epsilon = (1-p)\epsilon_1 + p\epsilon_2$ , then

$$\epsilon_2 = \frac{\epsilon - (1-p)\epsilon_1}{p} \leq \frac{\epsilon}{p}. \quad (9)$$

Therefore,

$$\epsilon_2 < \max\left(\frac{\epsilon}{1-p}, \frac{\epsilon}{p}\right) = \frac{\epsilon}{p^*} \quad (10)$$

where  $p^* = \min\{p, 1-p\}$ .

## Appendix B. Proof for optimality of class-balanced sampling ( $p = 0.5$ )

Consider a two-class classification problem where the two classes are represented by  $A$  and  $B$ . Without loss of generality, let  $A$  be the minority class containing  $0 < \beta \leq 0.5$  fraction of the points. Let  $\hat{A}$  and  $\hat{B}$  represent the two classes after one or both the classes are corrupted with adversarial noise. Let  $\rho\beta$ ,  $0 \leq \rho < 1$  represent the upper limit on the fraction of corrupted points. Therefore, the total number of corrupted points can be written as  $n_c = \rho\beta\ell$ . Further, let  $\alpha$  be the fraction of the corrupted points that were originally in class  $B$  but were assigned to class  $A$ . Therefore, the fraction of the new classes can be given by

$$|\hat{A}| = \beta + \alpha\rho\beta - (1-\alpha)\rho\beta \quad (11)$$

$$|\hat{B}| = 1 - \beta - \alpha\rho\beta + (1-\alpha)\rho\beta \quad (12)$$

Moreover, the fraction of good (clean) and bad (misabeled) points in both the classes are

$$|\hat{A}_g| = \beta - (1-\alpha)\rho\beta \quad (13)$$

$$|\hat{A}_b| = \alpha\rho\beta \quad (14)$$

$$|\hat{B}_g| = 1 - \beta - \alpha\rho\beta \quad (15)$$

$$|\hat{B}_b| = (1-\alpha)\rho\beta \quad (16)$$

Therefore, the conditional probability of picking a good or a bad point for both the classes are given by

$$P(a_g|\hat{A}) = \frac{|\hat{A}_g|}{|\hat{A}|} = \frac{1 - (1 - \alpha)\rho}{1 + \alpha\rho - (1 - \alpha)\rho} \quad (17)$$

$$P(a_b|\hat{A}) = \frac{|\hat{A}_b|}{|\hat{A}|} = \frac{\alpha\rho}{1 + \alpha\rho - (1 - \alpha)\rho} \quad (18)$$

$$P(b_g|\hat{B}) = \frac{|\hat{B}_g|}{|\hat{B}|} = \frac{1 - \beta - \alpha\rho\beta}{1 - \beta - \alpha\rho\beta + (1 - \alpha)\rho\beta} \quad (19)$$

$$P(b_b|\hat{B}) = \frac{|\hat{B}_b|}{|\hat{B}|} = \frac{(1 - \alpha)\rho\beta}{1 - \beta - \alpha\rho\beta + (1 - \alpha)\rho\beta} \quad (20)$$

Assuming that the probability with which points from classes  $\hat{A}$  and  $\hat{B}$  are picked is given by  $P(\hat{A}) = p$  and  $P(\hat{B}) = 1 - p$  respectively, the probability of picking up a good or a bad point for both the classes are respectively given by  $P(a_g) = P(a_g|\hat{A})p$ ,  $P(a_b) = P(a_b|\hat{A})p$ ,  $P(b_g) = P(b_g|\hat{B})(1 - p)$  and  $P(b_b) = P(b_b|\hat{B})(1 - p)$ .

The probability  $\eta$  of not picking  $r/2$  clean points from either class is upper bounded by

$$\eta \leq \sum_{k=0}^{r/2-1} \binom{s}{k} (P(a_g))^k (1 - P(a_g))^{s-k} + \sum_{k=0}^{r/2-1} \binom{s}{k} (P(b_g))^k (1 - P(b_g))^{s-k}. \quad (21)$$

For worst case analysis, we need to maximize  $\eta$  and therefore, minimize both  $P(a_g)$  and  $P(b_g)$ , which in turn requires minimizing  $P(a_g|\hat{A})$  and  $P(b_g|\hat{B})$  w.r.t. both  $\alpha$  and  $\beta$ . Differentiating  $P(a_g|\hat{A})$  w.r.t  $\alpha$

$$\frac{dP(a_g|\hat{A})}{d\alpha} = \frac{\rho(-1 + \rho)}{(1 - \rho + 2\alpha\rho)^2} \leq 0. \quad (22)$$

Therefore,

$$\arg \min_{\alpha} P(a_g) = 1. \quad (23)$$

Similarly, differentiating  $P(b_g|\hat{B})$  w.r.t.  $\alpha$

$$\frac{dP(b_g|\hat{B})}{d\alpha} = \frac{\rho\beta(1 - \beta(1 + \rho))}{(1 - \beta + \rho\beta - 2\alpha\rho\beta)^2} \geq 0 \quad (24)$$

Therefore,

$$\arg \min_{\alpha} P(b_g) = 0. \quad (25)$$

Also,

$$\frac{dP(b_g|\hat{B})}{d\alpha} = \frac{-\rho(1 - \alpha)}{(1 - \beta + \rho\beta - 2\alpha\rho\beta)^2} \leq 0 \quad (26)$$

implying that

$$\arg \min_{\beta} P(b_g) = \frac{1}{2}. \quad (27)$$

Substituting  $\alpha = 1$  in  $P(a_g)$  and  $\alpha = 0, \beta = 1/2$  in  $P(b_g)$ , we get

$$\min P(a_g) = \frac{p}{1+\rho} \quad \text{and} \quad \min P(b_g) = \frac{1-p}{1+\rho}. \quad (28)$$

Therefore, the worst case bound for (21) can be written as

$$\eta \leq \sum_{k=0}^{r/2-1} \binom{s}{k} \left(\frac{p}{1+\rho}\right)^k \left(1 - \frac{p}{1+\rho}\right)^{s-k} + \sum_{k=0}^{r/2-1} \binom{s}{k} \left(\frac{1-p}{1+\rho}\right)^k \left(1 - \frac{1-p}{1+\rho}\right)^{s-k}. \quad (29)$$

Applying Hoeffding bound Hoeffding (1963) individually on each of the two terms

$$\eta \leq \frac{1}{2} \exp\left(-\frac{2}{s} \left(\frac{sp}{1+\rho} - \frac{r}{2} + 1\right)^2\right) + \frac{1}{2} \exp\left(-\frac{2}{s} \left(\frac{s(1-p)}{1+\rho} - \frac{r}{2} + 1\right)^2\right) \quad (30)$$

as long as  $\frac{sp}{1+\rho} > \frac{r}{2} - 1$  and  $\frac{s(1-p)}{1+\rho} > \frac{r}{2} - 1$ . The RHS of 30 can be rewritten as

$$f = \frac{1}{2} \exp\left(-\frac{1}{2} \left(\frac{p - \frac{(r-2)(1+\rho)}{2s}}{\left(\frac{1+\rho}{2\sqrt{s}}\right)}\right)^2\right) + \frac{1}{2} \exp\left(-\frac{1}{2} \left(\frac{(1-p) - \frac{(r-2)(1+\rho)}{2s}}{\left(\frac{1+\rho}{2\sqrt{s}}\right)}\right)^2\right) \quad (31)$$

which is simply the sum of two Gaussians with means  $\mu_1 = \frac{(r-2)(1+\rho)}{2s}$  and  $\mu_2 = 1 - \frac{(r-2)(1+\rho)}{2s}$  and equal variance  $\sigma = \frac{1+\rho}{2\sqrt{s}}$ . Differentiating the above expression w.r.t.  $p$

$$\frac{df}{dp} = -\frac{2 \left(\frac{sp}{1+\rho} - \frac{r}{2} + 1\right)}{\exp\left(\frac{2}{s} \left(\frac{sp}{1+\rho} - \frac{r}{2} + 1\right)^2\right) (1+\rho)} + \frac{2 \left(\frac{s(1-p)}{1+\rho} - \frac{r}{2} + 1\right)}{\exp\left(\frac{2}{s} \left(\frac{s(1-p)}{1+\rho} - \frac{r}{2} + 1\right)^2\right) (1+\rho)}. \quad (32)$$

It can be clearly seen that  $p = 0.5$  is a solution of (32). Also, the sum of two Gaussians can be either unimodal ( $p = 0.5$  is global maximum) or bimodal ( $p = 0.5$  is a minimum) Behboodian (1970). The second order derivative of  $f$  w.r.t.  $p$  can be written as

$$\frac{d^2f}{dp^2} = \frac{8 \left(\frac{sp}{1+\rho} - \frac{r}{2} + 1\right)^2 - 2s}{\exp\left(\frac{2}{s} \left(\frac{sp}{1+\rho} - \frac{r}{2} + 1\right)^2\right) (1+\rho)^2} + \frac{8 \left(\frac{s(1-p)}{1+\rho} - \frac{r}{2} + 1\right)^2 - 2s}{\exp\left(\frac{2}{s} \left(\frac{s(1-p)}{1+\rho} - \frac{r}{2} + 1\right)^2\right) (1+\rho)^2}. \quad (33)$$

Therefore, enforcing a minimum at  $p = 0.5$ , we get the condition that

$$\left.\frac{d^2f}{dp^2}\right|_{p=0.5} = \frac{32 \left(\frac{s}{2(1+\rho)} - \frac{r}{2} + 1\right)^2}{\exp\left(\frac{2}{s} \left(\frac{s}{2(1+\rho)} - \frac{r}{2} + 1\right)^2\right) (1+\rho)^2} - \frac{8s}{\exp\left(\frac{2}{s} \left(\frac{s}{2(1+\rho)} - \frac{r}{2} + 1\right)^2\right) (1+\rho)^2} \geq 0. \quad (34)$$

This directly implies that

$$s \geq (\rho + 1) \left( r - 2 + \frac{1}{2}(\rho + 1) \left( 1 + \left( \frac{4r + \rho - 7}{\rho + 1} \right)^{1/2} \right) \right) \quad (35)$$



which, as expected, is a stronger condition than the one required for imposing the Hoeffding bound at  $p = 0.5$ , i.e.,  $s > (1 + \rho)(r - 2)$ . Furthermore, to enforce  $p = 0.5$  to be the *global* minimum, we impose the condition that the value of  $f$  at  $p = 0.5$  is strictly less than that at any of the two extreme points of  $f$  (i.e., at  $p = \frac{1}{s}(1 + \rho)(r/2 - 1)$  and  $p = 1 - \frac{1}{s}(1 + \rho)(r/2 - 1)$ ). This gives us an even stronger condition

$$s \geq (\rho + 1) \left( r - 2 + (\rho + 1) \left( \log 2 + \left( \frac{\log 2 (\log 2 + 2r + \rho \log 2 - 4)}{\rho + 1} \right)^{1/2} \right) \right). \quad (36)$$

This is the sufficient condition to guarantee that the worst case probability of selecting less than  $r/2$  clean points per class is minimum at  $p = 0.5$ , i.e., when *class-balanced* sampling is performed over the data.

### Appendix C. Details of performance metrics

For class-imbalanced data sets, very high classification accuracy can be trivially obtained by labeling the entire data with the majority class label. The use of Balanced Accuracy (BAC) for class-imbalanced data sets is prescribed by Brodersen et al. (2010) and can be simply computed as

$$\text{BAC} = \frac{\text{sensitivity} + \text{specificity}}{2}. \quad (37)$$

The *sensitivity* and *specificity* are defined as follows

$$\text{sensitivity} = \frac{tp}{tp + fn} \quad (38)$$

$$\text{specificity} = \frac{tn}{tp + fn} \quad (39)$$

where  $tp$  and  $fp$  denote the number of true and false positives while  $tn$  and  $fn$  denote the number of true and false negatives.

Similarly, traditional F-score can be trivially maximized for imbalanced data sets by compromising recall for high precision. Therefore, SIF Flach (2003) serves as an alternative to the F-score for imbalanced data sets and is given by

$$\text{SIF} = \frac{2tpr}{tpr + fpr + 1} \quad (40)$$

where  $tpr$  and  $fpr$  are true and false positive rates respectively. Like BAC, SIF also reduces to traditional F-score for class-balanced data sets. Another popular metric for comparison of classification performances is Area Under the ROC Curve (AUC). Although, unlike BAC and SIF, AUC is not a skew-insensitive measure, we also computed AUC measures for all the methods. It is important to mention that **SubSVMs** is always comparable to that of the other methods w.r.t. AUC. Finally, we note that for the results reported using AUC, we needed to retrain an SVM on the error-corrected data (unlike earlier, when we directly used majority voting on the test data).

	<i>Method</i>	a1a	a2a	a3a	a4a	a5a	sp1c	mush	svm1	w1a	w2a	w3a	w4a	w5a
<i>clean</i>	Oracle-SVM	0.71	0.73	0.71	0.72	0.72	0.91	1.00	0.97	0.66	0.69	0.72	0.74	0.76
	Blind-SVM	0.71	0.72	0.71	0.71	0.71	0.89	1.00	0.97	0.66	0.68	0.72	0.74	0.76
	Bag-SVM	0.71	0.73	0.72	0.72	0.72	0.90	1.00	0.97	0.63	0.67	0.69	0.73	0.74
	CV-SVM	0.71	0.72	0.71	0.71	0.72	0.90	1.00	0.97	0.55	0.56	0.55	0.61	0.64
	SubSVMs	<b>0.81</b>	<b>0.82</b>	<b>0.82</b>	<b>0.82</b>	<b>0.82</b>	0.87	0.98	0.96	<b>0.85</b>	<b>0.86</b>	<b>0.86</b>	<b>0.88</b>	<b>0.88</b>
$\alpha = 1.0$	Oracle-SVM	0.79	0.80	0.80	0.81	0.81	0.70	0.74	0.90	0.66	0.70	0.72	0.74	0.76
	Blind-SVM	0.74	0.75	0.75	0.77	0.77	0.70	0.68	0.87	0.66	0.69	0.70	0.74	0.75
	Bag-SVM	0.79	0.80	0.80	0.80	0.81	0.69	0.73	0.90	0.64	0.69	0.71	0.74	0.75
	CV-SVM	0.77	0.79	0.79	0.79	0.79	0.69	0.69	0.90	0.64	0.59	0.60	0.67	0.70
	SubSVMs	0.81	0.81	0.81	0.82	0.82	<b>0.81</b>	<b>0.97</b>	0.91	<b>0.79</b>	<b>0.84</b>	<b>0.84</b>	<b>0.85</b>	<b>0.86</b>
$\alpha = 0.5$	Oracle-SVM	0.51	0.52	0.52	0.52	0.52	0.72	0.99	0.89	0.45	0.48	0.51	0.53	0.55
	Blind-SVM	0.51	0.52	0.52	0.52	0.52	0.64	0.65	0.89	0.41	0.47	0.51	0.53	0.55
	Bag-SVM	0.50	0.51	0.51	0.51	0.51	0.72	0.99	0.89	0.41	0.43	0.48	0.50	0.53
	CV-SVM	0.25	0.21	0.31	0.34	0.34	0.71	0.96	0.71	0.32	0.23	0.29	0.31	0.29
	SubSVMs	<b>0.80</b>	<b>0.80</b>	<b>0.81</b>	<b>0.81</b>	<b>0.82</b>	0.75	0.98	<b>0.94</b>	<b>0.78</b>	<b>0.81</b>	<b>0.81</b>	<b>0.84</b>	<b>0.85</b>
$\alpha = 0.0$	Oracle-SVM	0.22	0.22	0.23	0.22	0.22	0.20	0.37	0.01	0.16	0.17	0.19	0.18	0.18
	Blind-SVM	0.22	0.22	0.23	0.22	0.22	0.13	0.37	0.01	0.14	0.17	0.18	0.18	0.18
	Bag-SVM	0.14	0.15	0.15	0.16	0.16	0.15	0.31	0.01	0.12	0.14	0.15	0.14	0.15
	CV-SVM	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.12	0.07	0.07	0.03
	SubSVMs	<b>0.78</b>	<b>0.78</b>	<b>0.79</b>	<b>0.79</b>	<b>0.81</b>	<b>0.77</b>	<b>0.97</b>	<b>0.96</b>	<b>0.69</b>	<b>0.75</b>	<b>0.80</b>	<b>0.80</b>	<b>0.81</b>

Table 4: Skew-Insensitive F-score (SIF) under L-2 loss for clean and noisy versions of UCI data sets. For different types of attacks (different  $\alpha$ ) the results for each method are averaged over 10 different noisy versions. ‘sp1c’, ‘mush’ and ‘svm1’ stand for splice, mushrooms and svmguide1. Only CV-SVM and SubSVMs are agnostic to the true test labels. The cases where one of the methods is significantly better than all others ( $\geq 0.05$ ) are highlighted.

## Appendix D. Additional Results

Tables 4, 5 and 6 present additional results on the UCI data sets under L-2 loss using Skew-Insensitive F-Score (SIF), Area Under the Curve (AUC) and Accuracy, respectively. Table 7 shows detailed run-times corresponding to Table 3 in the paper.

## References

- R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *ECML*, pages 39–50, 2004.
- Javad Behboodian. On the modes of a mixture of two normal distributions. *Technometrics*, 12(1):131–139, 1970.

	<i>Method</i>	a1a	a2a	a3a	a4a	a5a	splc	mush	svm1	w1a	w2a	w3a	w4a	w5a
<i>clean</i>	Oracle-SVM	0.89	0.90	0.90	0.90	0.90	0.96	1.00	1.00	0.93	0.95	0.96	0.96	0.96
	Blind-SVM	0.89	0.90	0.90	0.90	0.90	0.95	1.00	0.99	0.93	0.95	0.95	0.96	0.96
	Bag-SVM	0.88	0.89	0.89	0.89	0.89	0.96	1.00	1.00	0.80	0.88	0.90	0.91	0.94
	CV-SVM	0.89	0.90	0.90	0.90	0.90	0.96	1.00	1.00	0.91	0.95	0.96	0.96	0.96
	SubSVMs	0.89	0.89	0.89	0.90	0.90	0.93	0.98	0.99	0.90	0.92	0.92	0.93	0.94
$\alpha = 1.0$	Oracle-SVM	0.88	0.89	0.89	0.89	0.89	0.88	1.00	0.99	0.92	0.93	0.94	0.94	0.95
	Blind-SVM	0.88	0.89	0.89	0.89	0.89	0.85	1.00	0.99	0.91	0.93	0.94	0.94	0.94
	Bag-SVM	0.87	0.88	0.89	0.89	0.89	0.88	0.56	0.99	0.98	0.92	0.88	0.92	0.90
	CV-SVM	0.86	0.87	0.87	0.88	0.88	0.88	0.99	0.99	0.89	0.92	0.94	0.94	0.94
	SubSVMs	0.88	0.88	0.89	0.89	0.89	0.88	0.98	0.99	0.86	0.90	0.91	0.92	0.93
$\alpha = 0.5$	Oracle-SVM	0.88	0.88	0.88	0.88	0.89	0.82	1.00	0.99	0.91	0.92	0.93	0.94	0.94
	Blind-SVM	0.88	0.88	0.88	0.88	0.89	0.82	1.00	0.96	0.91	0.92	0.92	0.93	0.94
	Bag-SVM	0.82	0.84	0.85	0.86	0.86	0.85	1.00	0.99	0.98	0.97	0.89	0.89	0.86
	CV-SVM	0.84	0.84	0.86	0.86	0.88	0.78	0.97	0.99	0.89	0.91	0.92	0.93	0.93
	SubSVMs	0.87	0.87	0.88	0.88	0.89	0.82	0.99	0.99	0.86	0.88	0.89	0.91	0.92
$\alpha = 0.0$	Oracle-SVM	0.87	0.88	0.88	0.89	0.89	0.87	1.00	0.99	0.89	0.90	0.92	0.93	0.93
	Blind-SVM	0.87	0.88	0.88	0.89	0.89	0.87	1.00	0.98	0.88	0.90	0.92	0.93	0.93
	Bag-SVM	0.50	0.50	0.50	0.50	0.50	0.53	0.08	0.94	0.98	0.98	0.98	0.98	0.98
	CV-SVM	0.86	0.86	0.88	0.88	0.89	0.86	1.00	0.99	0.87	0.88	0.91	0.92	0.92
	SubSVMs	0.87	0.88	0.88	0.89	0.89	0.88	0.99	0.99	0.83	0.86	0.88	0.90	0.91

Table 5: Area Under the Curve (AUC) under L-2 loss for clean and noisy versions of UCI data sets. For different types of attacks (different  $\alpha$ ) the results for each method are averaged over 10 different noisy versions. ‘splc’, ‘mush’ and ‘svm1’ stand for splice, mushrooms and svmguide1. Only CV-SVM and SubSVMs are agnostic to the true test labels. The cases where one of the methods is significantly better than all others ( $\geq 0.05$ ) are highlighted.

	<i>Method</i>	a1a	a2a	a3a	a4a	a5a	splc	mush	svm1	w1a	w2a	w3a	w4a	w5a
<i>clean</i>	Oracle-SVM	0.84	0.85	0.85	0.85	0.85	0.91	1.00	0.97	0.98	0.98	0.98	0.98	0.99
	Blind-SVM	0.84	0.84	0.84	0.85	0.85	0.90	1.00	0.96	0.98	0.98	0.98	0.98	0.98
	Bag-SVM	0.84	0.85	0.85	0.85	0.85	0.90	1.00	0.97	0.98	0.98	0.98	0.98	0.99
	CV-SVM	0.84	0.84	0.85	0.85	0.85	0.91	1.00	0.97	0.98	0.98	0.98	0.98	0.99
	SubSVMs	0.78	0.78	0.78	0.79	0.79	0.86	0.98	0.96	0.84	0.84	0.83	0.85	0.86
$\alpha = 1.0$	Oracle-SVM	0.80	0.79	0.81	0.81	0.81	0.56	0.63	0.89	0.98	0.98	0.98	0.98	0.99
	Blind-SVM	0.79	0.78	0.80	0.81	0.81	0.51	0.49	0.82	0.98	0.98	0.98	0.98	0.98
	Bag-SVM	0.80	0.79	0.81	0.81	0.81	0.53	0.61	0.89	0.98	0.98	0.98	0.98	0.99
	CV-SVM	0.77	0.77	0.78	0.80	0.79	0.48	0.49	0.89	0.98	0.97	0.98	0.98	0.98
	SubSVMs	0.73	0.73	0.74	0.74	0.74	<b>0.77</b>	<b>0.97</b>	0.90	0.78	0.76	0.77	0.82	0.81
$\alpha = 0.5$	Oracle-SVM	0.80	0.80	0.81	0.81	0.81	0.74	0.99	0.90	0.98	0.98	0.98	0.98	0.98
	Blind-SVM	0.80	0.80	0.81	0.81	0.81	0.70	0.95	0.85	0.97	0.97	0.97	0.97	0.97
	Bag-SVM	0.80	0.80	0.81	0.81	0.81	0.74	0.99	0.90	0.98	0.98	0.98	0.98	0.98
	CV-SVM	0.79	0.80	0.80	0.80	0.80	0.71	0.95	0.90	0.97	0.97	0.98	0.98	0.98
	SubSVMs	0.75	0.75	0.75	0.76	0.76	0.74	0.97	0.93	0.75	0.75	0.81	0.82	0.83
$\alpha = 0.0$	Oracle-SVM	0.77	0.77	0.77	0.77	0.77	0.57	0.63	0.50	0.97	0.97	0.97	0.97	0.97
	Blind-SVM	0.77	0.77	0.77	0.77	0.77	0.55	0.63	0.50	0.97	0.97	0.97	0.97	0.97
	Bag-SVM	0.77	0.77	0.77	0.77	0.77	0.56	0.61	0.50	0.97	0.97	0.97	0.97	0.97
	CV-SVM	0.76	0.76	0.76	0.76	0.76	0.52	0.52	0.50	0.97	0.97	0.97	0.97	0.97
	SubSVMs	0.80	0.79	0.79	0.80	0.80	<b>0.79</b>	<b>0.97</b>	<b>0.96</b>	0.86	0.84	0.82	0.87	0.93

Table 6: Accuracy under L-2 loss for clean and noisy versions of UCI data sets. For different types of attacks (different  $\alpha$ ) the results for each method are averaged over 10 different noisy versions. ‘splc’, ‘mush’ and ‘svm1’ stand for splice, mushrooms and svmguide1. Only CV-SVM and SubSVMs are agnostic to the true test labels. The cases where one of the methods is significantly better than all others ( $\geq 0.05$ ) are highlighted.

	<i>Method</i>	a1a	a2a	a3a	a4a	a5a	splc	mush	svm1	w1a	w2a	w3a	w4a	w5a
<i>clean</i>	Oracle-SVM	180	266	376	601	937	38	102	19	213	286	372	532	805
	Blind-SVM	175	258	360	588	917	38	81	17	207	269	347	483	746
	Bag-SVM	64	134	276	617	1675	64	167	38	26	51	103	215	369
	CV-SVM	138	286	579	1465	2860	190	1033	102	82	160	293	654	1146
	SubSVMs	3	3	3	4	4	4	7	3	3	3	3	4	5
$\alpha = 1.0$	Oracle-SVM	291	418	606	957	2211	47	372	61	387	568	840	1521	2528
	Blind-SVM	292	420	610	960	2213	47	370	59	383	568	840	1520	2527
	Bag-SVM	73	147	300	703	3272	69	1179	358	59	129	224	710	1951
	CV-SVM	242	499	1034	2616	5036	214	3143	532	157	329	678	1633	3029
	SubSVMs	4	5	5	6	7	5	8	4	4	4	5	6	7
$\alpha = 0.5$	Oracle-SVM	281	408	588	931	1980	55	1513	67	304	460	639	1144	1905
	Blind-SVM	280	408	590	938	1980	55	1507	67	304	459	640	1144	1908
	Bag-SVM	100	256	518	860	3859	72	3937	330	39	76	146	365	828
	CV-SVM	224	465	936	2364	4598	262	6597	659	108	229	443	1095	2070
	SubSVMs	4	5	5	6	7	5	8	4	4	4	5	6	7
$\alpha = 0.0$	Oracle-SVM	118	180	256	393	598	31	257	28	120	181	237	347	495
	Blind-SVM	118	171	240	385	571	31	273	22	123	176	229	329	482
	Bag-SVM	21	42	80	197	342	46	899	105	14	29	54	112	206
	CV-SVM	75	154	311	771	1510	141	2332	160	34	65	123	278	507
	SubSVMs	3	3	4	4	5	5	7	3	3	7	3	4	5

Table 7: Training times in seconds (rounded to the closest integer) for all the methods trained using L-2 loss averaged over 10 random label-manipulated versions of the data sets, corresponding to the results presented in Table 1 of the paper. ‘splc’, ‘mush’ and ‘svm1’ stand for splice, mushrooms and svmguide1. Note that for SubSVMs, the reported time is the time taken to train all the  $J = 1000$  SVMs on  $s = \log^2 \ell$ -size subsets.

- S. Ben-David, N. Eiron, and P.M. Long. On the difficulty of approximately maximizing agreements. *J. Comput. Syst. Sci.*, 66(3):496–514, 2003.
- A. Blum, A. Frieze, R. Kannan, and S. Vempala. A polynomial time algorithm for learning noisy linear threshold functions. In *FOCS*, pages 330–338, 1996.
- Leo Breiman. Heuristics of instability and equalization in model selection. *The annals of statistics*, 24(6):2350–2383, 1996.
- K.H. Brodersen, C.S. Ong, K.E. Stephan, and J.M. Buhmann. The balanced accuracy and its posterior distribution. In *ICPR*, pages 3121–3124, 2010.
- C.E. Brodley and M.A. Friedl. Identifying mislabelled training data. *J. AI Res.*, 11:131–167, 1999.
- P. Buhlmann and B. Yu. Analyzing bagging. *The annals of Statistics*, 30:927–961, 2002.
- A. Buja and W. Stuetzle. Smoothing effects of bagging. *Preprint, AT&T Labs Research*, 2000.
- C. Caramanis and S. Mannor. Learning in the limit with adversarial disturbances. In *COLT*, 2008.
- N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.
- O. Dekel and O. Shamir. Good learners for evil teachers. In *ICML*, pages 233–240, 2009.
- A. Elisseeff, T. Evgeniou, and M. Pontil. Stability of randomized learning algorithms. *JMLR*, 6:55–79, 2005.
- V. Feldman, P. Gopalan, S. Khot, and A.K. Ponnuswami. New results for learning noisy parities and halfspaces. In *FOCS*, 2006.
- P.A. Flach. The geometry of ROC space : Understanding machine learning metrics through ROC isometrics. In *ICML*, pages 194–201, 2003.
- Y. Grandvalet. Bagging equalized influence. *Machine Learning*, 55(3):251–270, 2004.
- V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *FOCS*, pages 543–552, 2006.
- S. Har-peled, D. Roth, and D. Zimak. Maximum margin coresets for active and noise tolerant learning. In *IJCAI*, 2006.
- J. Hastad. Some optimal inapproximability results. In *Theory of Computing*, pages 1–10, 1997.
- W. Hoeffding. Probability inequalities for sums of bounded random variables. *JASA*, 58:13–30, 1963.

- A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.
- M.J. Kearns, R.E. Schapire, and L.M. Sellie. Toward efficient agnostic learning. *ML*, 17: 115–141, 1994.
- P. Long and R. Servedio. Random classification noise defeats all convex potential boosters. *ML*, 78(3):287–304, 2010.
- P.M. Long and S.A. Servedio. Learning large-margin halfspaces with more malicious noise. In *NIPS*, pages 91–99, 2011.
- J. Saketha Nath and C. Bhattacharyya. Maximum margin classifiers with specified false positive and false negative error rates. In *SDM*, 2007.
- P.S. Sastry, G.D. Nagendra, and N. Manwani. A team of continuous-action learning automata for noise-tolerant learning of half-spaces. *IEEE TSMC*, 40(1):19–28, 2010.
- R. Servedio. Smooth boosting and learning with malicious noise. *JMLR*, 4:633–648, 2003.
- S. Shalev-Shwartz, O. Shamir, and K. Sridharan. Learning kernel-based halfspaces with the zero-one loss. In *COLT*, 2010.
- T.B. Trafalis and R.C. Gilbert. Maximum margin classifiers with noisy data: a robust optimization approach. In *IJCNN*, 2005.
- G. Valentini. Random aggregated and bagged ensembles of svms: An empirical bias-variance analysis. In F. Roli, J. Kittler, and T. Windeatt, editors, *LNCS, MCS 2004*, volume 3077, pages 263–272, 2004.
- L. Xu, K. Crammer, and D. Schuurmans. Robust support vector machine training via convex outlier ablation. In *AAAI*, 2006.